Hindawi
Complexity
Volume 2018, Article ID 4867607, 14 pages
https://doi.org/10.1155/2018/4867607

WILEY | Hindawi

Research Article

Application of the Variable Precision Rough Sets Model to Estimate the Outlier Probability of Each Element

Francisco Maciá Pérez,[1] **Jose Vicente Berna Martienz**,[1] **Alberto Fernández Oliva**,[2] **and Miguel Abreu Ortega**[3]

[1]*Computer Science & Technology Department, University of Alicante, Spain*
[2]*Department of Computer Science, Faculty of Mathematics and Computer Science, University of Havana, Cuba*
[3]*MSc Student at Georgia Institute of Technology, USA*

Correspondence should be addressed to Jose Vicente Berna Martienz; jvberna@ua.es

Received 26 April 2018; Accepted 2 September 2018; Published 8 October 2018

Guest Editor: Magnus Johnsson

In a data mining process, outlier detection aims to use the high marginality of these elements to identify them by measuring their degree of deviation from representative patterns, thereby yielding relevant knowledge. Whereas rough sets (RS) theory has been applied to the field of knowledge discovery in databases (KDD) since its formulation in the 1980s; in recent years, outlier detection has been increasingly regarded as a KDD process with its own usefulness. The application of RS theory as a basis to characterise and detect outliers is a novel approach with great theoretical relevance and practical applicability. However, algorithms whose spatial and temporal complexity allows their application to realistic scenarios involving vast amounts of data and requiring very fast responses are difficult to develop. This study presents a theoretical framework based on a generalisation of RS theory, termed the variable precision rough sets model (VPRS), which allows the establishment of a stochastic approach to solving the problem of assessing whether a given element is an outlier within a specific universe of data. An algorithm derived from quasi-linearisation is developed based on this theoretical framework, thus enabling its application to large volumes of data. The experiments conducted demonstrate the feasibility of the proposed algorithm, whose usefulness is contextualised by comparison to different algorithms analysed in the literature.

1. Introduction

Outlier detection is an area of increasing relevance within the more general data mining process. Outliers may highlight extremely important findings in a wide range of applications: fraud detection, detection of illegal access to corporate networks, and detection of errors in input data, among others.

The rough sets basic model created by Pawlak [1] is a model with a simple and solid mathematical basis: the equivalence relation theory, which enables the description of partitions consisting of classes of indiscernible objects. The rough sets (RS) rationale consists of approximating a set using a pair of sets, termed lower and upper approximations. In general, the RS approach is based on the ability to classify data collected through various means. In recent years, this model has been successfully applied in various contexts [2–4]. Therefore, its study has attracted the attention of the international scientific community, especially regarding solving problems that involve establishing relationships between data.

An outlier detection method is proposed in [5], which is the first Pawlak rough sets application to this problem. However, its computational implementation is complicated by its exponential order. An extension of the theoretical framework of the previous proposition is presented in [6], in which an outlier detection algorithm is implemented based on Pawlak rough sets—the Pawlak rough sets algorithm—with a nonexponential order of temporal and spatial complexity. In [6], a method for the detection of outliers has been proposed with a simple and rigorous theoretical setup, starting from a definition of outliers that is simple, intuitive, and computationally viable for large

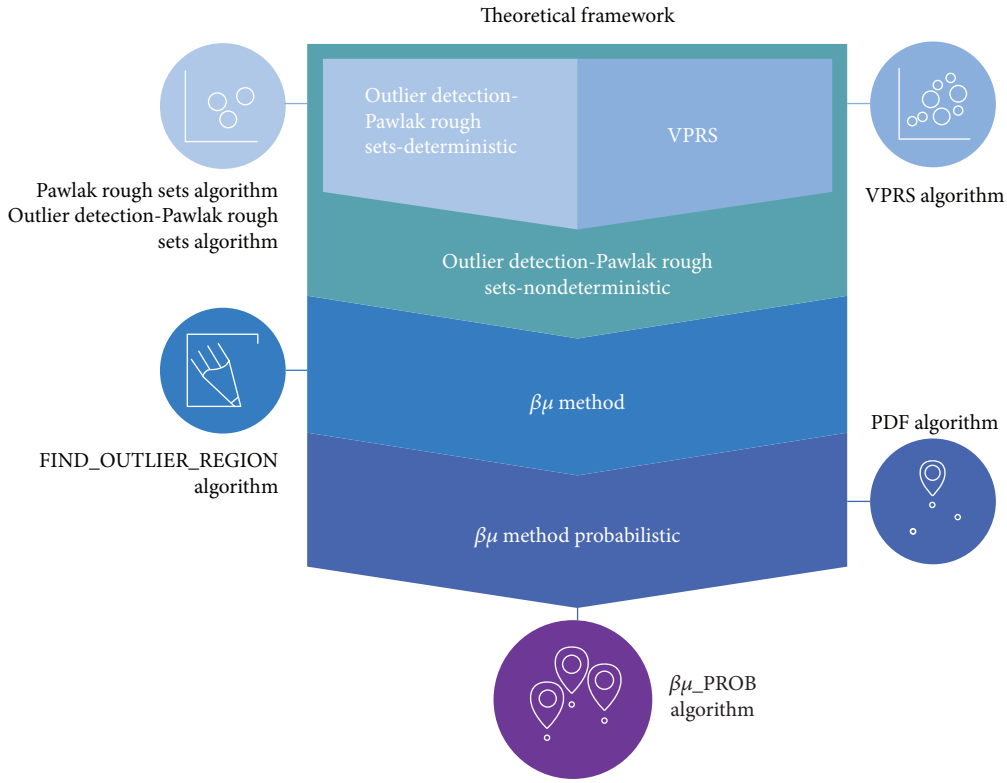

Figure 1: Global view of the theoretical framework.

datasets. From this method, an efficient algorithm for outlier mining has been developed, conceptually based on a novel and original approach using rough set theory, which has not been applied in any previous category of classification for the methods of rough set detection. The proposed algorithm is linear with respect to the cardinality of the data universe over which it is applied, and it is quadratic with respect to the number of equivalence relations used to describe the universe. However, this number of relations merely represents a constant, as it is usually significantly smaller than the cardinality of the universe in question. In contrast to many other methods that present difficulties in their application depending on the nature of the data to be analyzed, our proposal is applicable to both continuous and discrete data. The possibility that the datasets may contain a mix of attribute types (e.g., a mix of continuous and categorical attributes) does not present a limitation for the applicability of the proposed algorithm. Nevertheless, this result has the drawback for our purposes of inheriting the deterministic nature of the Pawlak rough sets regarding the classification.

The variable precision rough sets model (VPRS) [7] is a generalisation of the Pawlak rough sets that rectifies its deterministic nature through a new concept of inclusion of standard sets: the inclusion of majority sets [8, 9], which makes it possible to incorporate user-defined thresholds. A computationally viable algorithm for the nondeterministic detection of outliers, termed the VPRS algorithm, based on the VPRS, which was in turn based on the theoretical framework provided by Pawlak rough sets and VPRS, termed *nondeterministic outlier detection-Pawlak rough sets* (Figure 1), is presented in [10]. Figure 1 shows a global view of the theoretical framework for the formalisation of a computationally viable algorithm for unsupervised probabilistic estimation of the outlier condition of each element of a given universe of data used in this paper.

The Pawlak rough sets and VPRS algorithms solve the following problem: "to determine the set of outliers of a given universe of data from a preset exceptionality threshold (μ) defined in [6] at a given allowed classification error (β) defined by [7]."

In this paper, a new approach to the problem of outlier detection that solves the limitations of the aforementioned results is proposed: to preset the thresholds and to develop scalable algorithms independent of the context and nature of the problem. Therefore, the aim of this research may be summarised as follows: "to create a computationally viable method that calculates the outlier probability of each element from a given universe of data without the need to establish preconditions—that is, the determination of the thresholds (μ, β) of the analysis—that depend on each specific context to which the algorithm is applied."

The starting hypothesis is summarised as follows: "a new theory may be developed by extending the basic concepts and the formal tools provided by RS theory [1, 11] and VPRS [7], applied to the outlier detection problem, which allows the unsupervised determination, for each element of a universe of data, of the region of threshold values (μ, β) in which such element is an outlier." Based on this approach, which was termed the *$\beta\mu$ Method* (see Figure 1), "the outlier probability

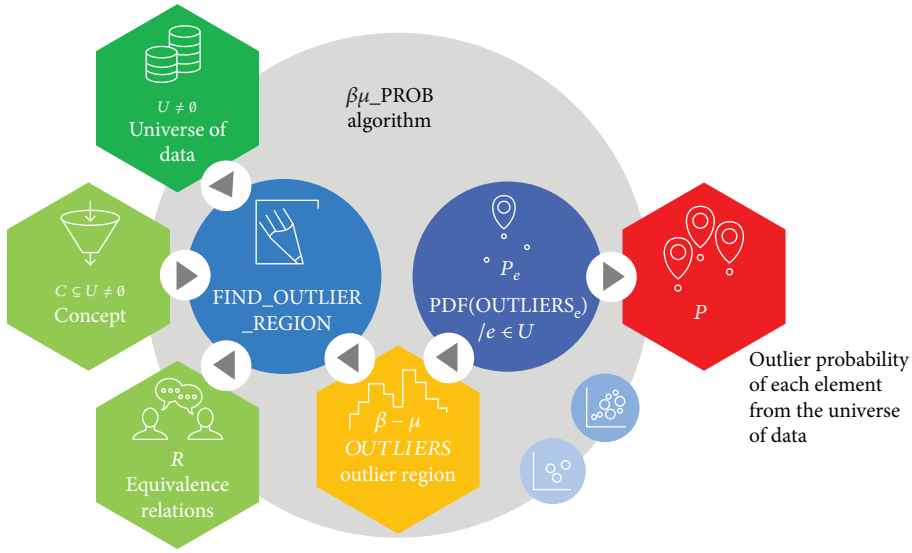

Figure 2: General outline of the proposed solution.

of each element from the universe of data can be determined." This new method is termed the *Probabilistic $\beta\mu$ Method* (see Figure 1).

To develop the method proposed in the research objective as a solution (see Figure 2), the theoretical framework developed in [6, 10] is expanded based on conceptual elements of the Pawlak rough sets and VPRS and on the theoretical proposition of [5]. Combined, they make it possible to formally demonstrate the theoretical elements proposed in the new concept of the method and serve as a reference framework to design and implement a computationally viable algorithm that validates the starting hypothesis. This algorithm has been termed the $\beta\mu$_PROB algorithm, as can be seen in Figure 2. This figure shows a general outline of the proposed solution, specified in the implementation of a computationally viable algorithm ($\beta\mu$_PROB Algorithm) for the unsupervised probabilistic estimation of the outlier condition of each element from a universe of data, entirely based on the development of the theoretical framework created in this research study.

Based on the above, the text below is divided into four sections. In Section 2, a theoretical framework termed *$\beta\mu$ Method* (Figure 1) is proposed alongside an algorithm that determines the outlier region of each element from the universe of data, termed the FIND_OUTLIER_REGION Algorithm (Figure 2). In Section 3, new theoretical elements collected using a method termed *Probabilistic $\beta\mu$ Method* (Figure 1) are proposed, and statistical techniques that make it possible to solve the problem posed are applied by proposing the $\beta\mu$_PROB algorithm (Figure 2), which determines the outlier probability of each element from the universe of data within such universe. In Section 4, the experiments that validate the proposed solution are designed, the findings are analysed, and the algorithms based on RS and the classical algorithms, in addition to the different RS algorithms that have been developed to achieve the final solution, are compared. In Section 5, the conclusions from this research study are presented, and some perspectives and future studies continuing this research are considered.

2. Outlier Region

In essence, the entire proposal in this article is summarized in the following two phases:

(i) In the first, it is determined for each element e of the finite universe U, under what conditions (threshold of exceptionality μ and classification error allowed β) that element behaves as an exceptional element (outlier). These conditions (μ and β) establish an R region within which the element is considered outlier

(ii) In the second phase, taking into account the determined R region, for each element of the finite universe U, the probability of each of them being an outlier in U is calculated using statistical techniques

To solve the problem, first, we expanded the theoretical framework defined in [6, 10] (Section 2.1). This framework is based on a method that we have termed the $\beta\mu$ Method. The method provides the formal tools that, second, make it possible to develop a computationally efficient algorithm to solve the problem, which we have termed the FIND_OUTLIER_REGION algorithm (Section 2.2).

2.1. Theoretical Framework: $\beta\mu$ Method. The $\beta\mu$ Method consists of three main tasks that can be easily differentiated: (a) to determine the outlier region in relation to threshold β, which makes it possible to calculate the allowable classification error, (b) to determine the outlier region in relation to threshold μ, that is, to calculate the preset outlier threshold, and (c) to integrate both specific solutions to determine the outlier region (β, μ) of each element from the universe of data. Below, we detail each of these tasks.

2.1.1. Outlier Region in Relation to β. To determine the outlier region in relation to the set of values of β (referred to as the allowable β-error in the classification), three specific subproblems are solved.

Subproblem 1: to determine the range of β values for which $B_i \subseteq B_j$, $i \neq j$, $1 \leq i, j \leq m$. B_i, B_j: *internal borders* with respect to equivalence relations i and j, where m is the total number of equivalence relations taken into account in the analysis. Based on the theoretical framework described in [6], it is known that if no internal border Bi is a subset of another internal border Bj, then all Bj elements are candidates for outliers in the dataset or universe of data, U. Therefore, the problem is restated as follows: to determine the set of β values for which an internal border Bi, $i \neq j$, is a subset of the internal border Bj, that is, $Bi \subseteq Bj$. After calculating this set, $\forall i \neq j$, $1 \leq i \leq m$, then the complement of the union of all ranges of β values calculated will be the set of values, in relation to such threshold, for which all Bj elements are candidates for outliers.

Subproblem 2: to determine the range of β values for which a given internal border is null. Similarly, in the theoretical framework on which the detection method is based, it is assumed that the internal borders considered in the analysis are not null. Accordingly, the β values for which this condition is met are determined. The analysis is performed for any internal border Bi, and subsequently, this result is generalised to any other internal border through a similar analysis.

Subproblem 3: to determine the set of β values for which $B_i = B_j$, $i \neq j$, $1 \leq i, j \leq m$. In the theoretical framework on which the detection method is based, the existence of two equal internal borders is not considered either, thereby requiring determining the set of β values for which this condition is met. In this case, the problem consists of determining the set of β values for which $B_i = B_j$, $i \neq j$, $1 \leq i, j \leq m$, which is easily deduced through the following sequence of equivalences: $B_i = B_j \Leftrightarrow B_i \subseteq B_j \wedge B_j \subseteq B_i \Leftrightarrow \beta \in I_{ij} \wedge \beta \in I_{ji} \Leftrightarrow \beta \in I_{ij} \cap I_{ji}$. From these, we can conclude that the set of β values for which $B_i = B_j$, $i \neq j$, $1 \leq i, j \leq m$, is $EQ_{ij} = \{\beta : \beta \in I_{ij} \cap I_{ji}\}$, in which I_{ij} is the set of β values for which $B_i \subseteq B_j$. $i \neq j$, $1 \leq i, j \leq m$.

After concluding the analysis of the three proposed subproblems, from the sequence of sets, a general criterion can be established defining when an internal border is a subset of another.

A: set of β values for which a nonempty internal border exits, which is a specific subset of the internal border j. $(I_{1j} - EQ_{1j} - N_1) \cup (I_{2j} - EQ_{2j} - N_2) \cup \cdots \cup (I_{mj} - EQ_{mj} - N_m) = A$, where N_i: set of β values for which $B_i = \phi$, $1 \leq i \leq m$.

A^c: set of β values for which no nonempty internal border is a specific subset of the internal border j.

S_j: set of β values for which no nonempty internal border is a specific subset of the internal border j excluding the values for which such border is empty. $S_j = A^c - N_j$.

Considering that for all B_j elements to be outliers, the condition that no other internal border is a subset of this border must be met; the previous results suggest that this only occurs when $\beta \in S_j$. Therefore, S_j is the range of β values for which an element e from the universe of data U, $e \in B_j$, belongs to some nonredundant outlier set, and thus e is a possible outlier.

2.1.2. Outlier Region in Relation to μ. The next step is to perform a similar analysis to determine the set of outlier threshold values μ for which each element from the universe of data may be considered an outlier. The problem is now the following: given an element $e \in U$, to determine the range of values of the threshold μ for which the outlier degree of e is higher than that of μ. The theoretical elements necessary to solve this problem are presented below according to the following logical sequence:

(i) To define the set of values of β for which $\forall e : e \in U$ belongs to internal border B_i, $1 \leq i \leq m$

(ii) To establish a new definition of outlier degree $\forall e : e \in U$, in a new interpretation of the values of β: ExcepDegree(e,β)

(iii) To determine $\forall e \in U$ the range of values of μ for which ExcepDegree(e,β) $\geq \mu$ for a given β value

Following this sequence, first, the set of β values for which $e \in U$ belongs to the internal border B_i, $1 \leq i \leq m$, is defined.

Definition 1. Let U be a universe of data, X the subset of values of U that meet a specific concept, $\forall e \in U$, $1 \leq i \leq m$, and EC an equivalence class of the partition induced by the equivalence relation r_i in U such that $e \in EC$. The set of values of β for which e belongs to the internal border B_i is defined as follows:

$$M_i(e) = \begin{cases} \beta : \beta < c(EC, X) < 1 - \beta, & \text{if } e \in X, \\ \varnothing, & \text{if } e \notin X, \end{cases} \quad (1)$$

wherein $c(A, B)$ is the measure of the degree of declassification of set A in relation to set B, that is, the relative error of classification of a set of objects, defined in the VPRS [7] as follows:

$$c(A, B) = \begin{cases} 1 - \dfrac{|A \cap B|}{|A|}, & \text{if } |A| \neq 0, \\ 0, & \text{if } |A| = 0. \end{cases} \quad (2)$$

As established by $M_i(e)$, the values of parameter β must meet the following restrictions to ensure that e belongs to the internal border B_i : $\beta < c(EC, X) < 1 - \beta \Rightarrow [\beta < 1 - c(EC, X)] \wedge [\beta < c(EC, X)]$. Therefore, the following range of β values within which $e \in B_i : \forall \beta : \beta \in [0, \min(-c(EC, X), 1 - c(EC, X))]$ can be established from $M_i(e)$. This result satisfies the criterion required to state that an

element $e \in U$ may be an outlier candidate. In this case, this means that it belongs to some internal border. Accordingly, below, a new definition of outlier degree of an element $e \in U$ is established, with a new interpretation: its dependence on the values of β. Preliminarily, a new definition and a new proposition must be established based on that dependence.

Definition 2. $\forall e \in U$, $1 \le i \le m$

$$\lambda_i(e) = \begin{cases} \text{Sup}(M_i(e)), & \text{if } M_i(\text{e}) \neq \varnothing, \\ 0, & \text{another case,} \end{cases} \tag{3}$$

wherein $\text{Sup}(M_i(e))$ is the lowest value of β that is higher than all values of the $M_i(e)$ range. For all $\beta < \lambda_i(e)$, the element e belongs to the internal border B_i. Thus,

Proposition 1. *$\forall e \in U$, $1 \le i \ne j \le m$, if $\lambda_i(e) \le \lambda_j(e) \Rightarrow \forall \beta$: $\beta < \lambda_i(e)$, $e \in B_i \wedge e \in B_j$. Based on the analysis performed, a specific sequence of the supremum $\lambda_i(e)$, $1 \le i \le m$ can be obtained for each element $e \in U$ associated with each internal border B_i, $Z_i(e)$. Being $Z_1(e), \ldots, Z_m(e)$, such that $\lambda_{Z1(e)}(e) \le \cdots \le \lambda_{Zm(e)}(e)$ a permutation of indices that order the $\lambda_i(e)$.*

Definition 3. With $e \in U$, $\beta \in [0\,;0,5)$ and m the number of internal borders considered in the analysis, the Total number of internal borders to which element e belongs at a given β value is defined as follows:

$$\text{Total}(e, \beta) = \begin{cases} m, & \text{if } \beta < \lambda_{Zl(e)}(e), \\ 0, & \text{if } \beta \ge \lambda_{Zm(e)}(e), \\ m - \max_k\left(\beta \ge \lambda_{Zk(e)}(e)\right), & \text{in another case.} \end{cases} \tag{4}$$

The first two parts of Definition 3 are established to ensure that when the max function is evaluated, a defined result is always established (especially when the condition established in the predicate $\lambda_{Zk(e)}(e)$ is not satisfied). The graphical interpretation of the $\text{Total}(e, \beta)$ function is illustrated in Figure 3. In this figure, $v = \max_k(\beta \ge \lambda_{Zk(e)}(e))$. This value is the highest value of k such that $(\beta \ge \lambda_{Zk(e)}(e))$, that is, is exactly the number of internal borders to which e does not belong. Furthermore, from $k' = k + 1$, $\beta < \lambda_{Zk}{}'(e)(e)$ will be fulfilled and therefore e belongs to the internal borders $B_{Zk}{}'(e), \ldots, B_{Zm(e)}$, by Proposition 1 and does not belong to the internal borders $B_{Z1(e)}, \ldots, B_{Zk(e)}$.

As a function of Definitions 2 and 3 and Proposition 1, the concept of the outlier degree of an element $e \in U$ is defined as a function of the β values.

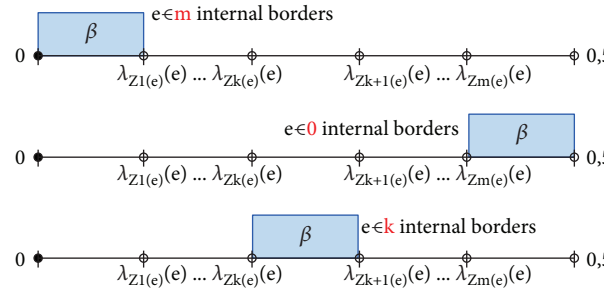

Figure 3: Graphic view of the $\text{Total}(e, \beta)$ function.

Definition 4. With $e \in U$ a value $\beta \in [0, 0.5]$ and m the number of internal borders considered in the analysis, the outlier degree of element e at a given β value is defined as follows: $\text{ExcepDegree}(e, \beta) = \text{Total}(e, \beta)/m$.

This definition does not contradict the proposition presented in [6]. Based on this proposition, $\forall e \in U$, the outlier degree of such element can be assessed for any β value and therefore the μ values for which $\text{ExcepDegree}(e,\beta) \ge \mu$.

2.1.3. Integrating Regions. The definitions above enable us to establish the following general method for determining the values of β and μ for which the element $e \in U$ is an outlier in U.

(1) To determine $M_i(e)$: β values for which the element $e \in B_i$

(2) To determine S_i: β values for which there is no internal border that is a subset of the internal border B_i

(3) To determine $D_i(e) = M_i(e) \cap S_i$: β values for which the element e belongs to B_i and there is no internal border that is a subset of the internal border B_i

For values of $\beta \in D_i(e)$, the element e belongs to some nonredundant outlier set and is the only representative of the internal border B_i in such set, that is, for β values in $D_i(e)$, $e \in Ei$

(4) $\forall \beta_o$, μ_o: $\beta_o \in \cup_{k=1}^{m} D_k(e) \wedge \mu_o \le \text{ExcepDegree}(e, \beta_o)$, then: e is an outlier in U. A $\beta_o \in \cup_{k=1}^{m} D_k(e)$ represents a value for which the element e belongs to some internal border of which no other internal border is a subset, and in such a case, μ_o must be lower than or equal to $\text{ExcepDegree}(e, \beta_o)$

Figure 4 shows the range of β-μ values for which any element e of the universe is an outlier in U. In this case, the following was assumed:

$$\text{range}(1) \cup \text{range}(2) = \cup_{k=1}^{m} D_k(e). \tag{5}$$

2.2. Computational Implementation: FIND_OUTLIER_REGION Algorithm. In this section, the FIND_OUTLIER_REGION algorithm is developed. This algorithm enables the unsupervised calculation of the range of values of the thresholds β-μ in which each element of the universe is an outlier.

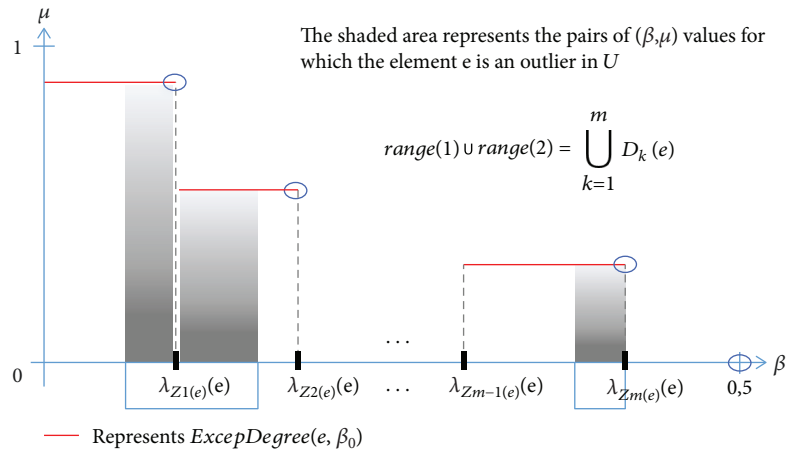

Figure 4: Range of β-μ values for which any element of the universe is an outlier in U.

This algorithm validates the β-μ method defined in the previous section and proceeds in three key steps.

(a) Calculation of the dependences between internal borders, or calculation of the inclusion relationship between them: BUILD_β_OUTILIER_REGION algorithm (see Algorithm 1)

(b) Calculation of the outlier region in relation to the threshold μ: BUILD_μ_OUTILIER_REGION algorithm (see Algorithm 2)

(c) Integration of both regions to obtain, for each element of the universe, the regions of β-μ values in which the element would be an outlier: OUTLIERS set and FIND_OUTLIER_REGIONS algorithm (see Algorithm 3)

All these algorithms contain the inputs *universe* U (dataset), $|U| = n$, and *concept* $X \subseteq U \neq \varnothing$ and the equivalence relationships $R = \{r_1, r_2, \ldots, r_n\}$.

The output of the *BUILD_β_OUTILIER_REGION* algorithm (Algorithm 1) is set S with the dependences between internal borders or the inclusion relationship between them. The output of the *BUILD_μ_OUTILIER_REGION* algorithm (Algorithm 2) consists of a tuple with two values: the outlier region ExcepDegree in relation to the outlier threshold μ and the set of classification errors β for which each element belongs to each equivalence relation $r_i \in R$.

Finally, the output of the *FIND_OUTLIER_REGION* algorithm (Algorithm 3) is the set of *OUTLIERS* with the regions of the β-μ values in which every element would be an outlier.

2.3. Analysis of the Complexity of the Method and the Algorithm. The temporal complexity of the algorithms depends on the number of ranges in the sets of specific ranges. Table 1 outlines the costs of each structure calculated for each algorithm. Based on these calculations, the temporal complexity of the FIND_OUTLIER_REGION algorithm is then determined, which, in the worst case, will be equal to the maximum of each of its three main tasks: $\mathbf{O}(n^2 \times m^2 \times \log(m))$.

The most original aspect of the *FIND_OUTLIER_REGION* algorithm is that it enables the unsupervised calculation of the range of threshold values (parameters β and μ) in which each element of the universe will be considered an outlier. However, the temporal and spatial complexity of the algorithm is of a higher order than that those of the algorithms Pawlak rough sets and VPRS [1, 7] because the result from the *FIND_OUTLIER_REGION* algorithm is more general.

When executing the algorithm once for a given data universe, the specific outputs of the previous algorithms can be obtained for any value of (β, μ). Determining, for each element of the universe, the total region of values of such thresholds in which such element is an outlier ensures that the entire universe can be subsequently searched for specific pairs of values of the thresholds (β, μ) belonging to the outlier region of any element. Thus, the usefulness of the FIND_OUTLIER_REGION algorithm becomes clear when seeking to assess the outlier condition of the elements of the universe for a given set of threshold values.

In summary, the result from the execution of the algorithm contains any particular result that could be obtained from the execution of the algorithms Pawlak rough sets and VPRS. This is the main advantage of the algorithm, compared with the expected advantage from increasing its temporal and spatial complexity when used only to calculate the regions of a single element of the universe.

Nevertheless, despite the high order of temporal complexity identified in the *worst case*, the algorithm can reach an order of temporal complexity similar to that of the algorithms Pawlak rough sets and VPRS, almost linear for the *best case* $\mathbf{\Omega}(n \times m^2 \times c)$.

The *OUTLIERS* region obtained allows a stochastic approximation to the solution of the problem of determining whether a given element is an outlier within a given universe of data (to establish a probabilistic criterion on such condition).

BUILD_β_OUTLIER_REGION (U, X, R): S

	Pseudo-code	*Comments*
1	**for each** r ∈ *R*	
2	**for each** q ∈ R − {r}	
3	S1[r][q] = {[0, 0.5)}	*Start solving Sub-problem No. 1*
4	S3[r][q] = {[0, 0.5)}	*Start solving Sub-problem No. 3*
5	S2[r] = {[0, 0.5)}	*Start solving Sub-problem No. 2*
6	**for each** r ∈ R	
7	P_r = CLASSIFY-ELEMENTS (U, r)	*Partition induced by the equiv. relation r*
8	class-max = 0	*starting the null minimum value [r]*
9	**for each** class ∈ P_r	
10	case1[r][class] = {[min(c(class, X), 1 - c(class, X)), 0.5)}	*Obtain the solution for the equivalence class for Case1*
11	class-max = max(class-max, c(class, X), 1 - c(class, X))	*Update the null minimum value[r]*
12	**for each** q ∈ R − {r}	*Searching the solution for the equiv. class of case2*
13	q-min = min(c(class, X), 1 - c(class, X))	*Minimum error of the equiv. classes according to q with elements of the equiv. class according to i*
14	**for each** e ∈ class	*For each class element*
15	q-class = CLASSIFY-ELEMENT(U, q, e)	*Obtain equiv. class to which it belongs according to q*
16	q-min = min(q-min, c(q-class, X), 1− c(q-class, X))	*Update the minimum value*
17	case2[r][q][class] = [0, q-min)}	*Obtain the solution of the equiv. class for Case 2*
18	S1[r][q] = S1[r][q] ∩ (case1[r][class] ∪ case2[r][q][class])	*Update S1 with new ranges of the equiv. class*
19	S2[r] = S2[r] ∩ {[class-max, 0.5)}	*Update S2 with new ranges of the equiv. class*
20	**for each** r ∈ R	*Update S3 from the S1 values*
21	**for each** q ∈ *R* − {r}	
22	**S3**[q][r] = **S1**[r][q] ∩ **S1**[q][r]	*Obtain the solution for which the internal border r is equal to q*
23	**for each** r ∈ R	*Calculate the outlier region for each internal border*
24	A = {}	*β for which the internal border r contains the other internal border*
25	**for each** q ∈ R − {r}	
26	A = A ∪ (**S1**[q][r]−**S3**[q][r]−**S2**[q])	*Update set A*
27	S[r] = {[0, 0.5)} - A − S2[r]	*Values for which the internal border r has no internal border*
28	**return** S	*Return the solution*

Algorithm 1: Pseudo-code of the *BUILD_β_OUTLIER_REGION* algorithm.

BUILD_μ_OUTLIER_REGION (U, X, R): {M, ExcepDegree}

	Pseudo-code	*Comments*
1	**for each** e ∈ U	*For each element of the universe*
2	**for each** r ∈ R	*For each equiv. relation*
3	class = CLASSIFY-ELEMENT(U, r, e) o	*Obtain the equiv. class of the element*
4	λ[e][r] = min(c(class, X), 1-c(class, X))	*Obtain the lowest β higher than all values of M[e][r]*
5	M[e][r] = {[0, λ[e][r])}	*Obtain the β for which the element belongs to r*
6	h = 1.0	
7	prev = 0.0	
8	**for each** inf ∈ SORT(λ[e])	*For each infimum in the order*
9	base = {}	*Obtain β ranges of height m*
10	ExcepDegree[e] = ExcepDegree [e] ∪ {[prev, inf) × [0, h]}	**Obtain the outlier rectangle**
11	prev = inf	**Save the value to form the next rectangle**
12	h = h − (1/|R|)	**Reduce the outlier rectangle height**
13	**return** <M, ExcepDegree>	**Return M and ExcepDegree**

Algorithm 2: Pseudo-code of the *BUILD_μ_OUTLIER_REGION* algorithm.

FIND_OUTLIER_REGION (U, X, R): OUTLIERS

	Pseudo-code	*Comments*
1	S = **BUILD_β_OUTLIER_REGION** (U, X, R)	*Step 1: calculation of the dependences between internal borders*
2	<M, ExcepDegree> = **BUILD_μ_OUTLIER_REGION** (U, X, R)	*Step 2: calculation of the outlier region*
		Integration of the regions
3	**for each** e ∈ U	*For each element of the universe*
4	D[e] = {}	
5	**for each** r ∈ R	*Values where e belongs to an internal border with no other internal border*
6	D[e] = D[e] ∪ M[e][r] ∩ S[r]	
7	OUTLIERS[e] = ExcepDegree[e] ∩ {D[e] × [0, 1]}	*Intersection between the outlier regions β and μ*
8	**return** OUTLIERS	*Return all regions*

Algorithm 3: Pseudo-code of the *FIND_OUTLIER_REGION* algorithm.

Table 1: Calculation of the spatial and temporal complexity of the FIND_OUTLIER_REGION algorithm by calculating the complexities of each structure of each component algorithm.

Algorithm	Data structure	Spatial complexity (*worst case*)	Temporal complexity (*worst case*)
BUILD_β_OUTLIER_REGION	Case1[i][ec]	$O(n \times m)$	$O(n \times m \times c)$
	Case2[i][j][ec]	$O(n \times m^2)$	$O(n \times m^2 \times c)$
	S1[i][j]	$O(n \times m^2)$	$O(n \times m \times \log(n))$
	S2[i]	$O(m)$	$O(n \times m)$
	S3[i][j]	$O(n \times m^2)$	$O(n \times m^2)$
	S[i]	$O(n \times m^2)$	$O(n \times m^2 \times \log(m))$
		$\boldsymbol{O(n^2 \times m^2)}$	$\boldsymbol{O(n \times m^2 \times \log(m))}$
BUILD_μ_OUTLIER_REGION	λ[e][i]	$O(n \times m)$	$O(n \times m \times c)$
	M[e][i]	$O(n \times m)$	$O(n \times m \times c)$
	ExcepDegree[e]	$O(n \times m)$	$O(n \times m \times \log(m))$
		$\boldsymbol{O(n \times m)}$	$\boldsymbol{O(n \times m \times \log(m))}$
FIND_OUTLIER_REGION	D[e]	$O(n^2 \times m^2)$	$O(n^2 \times m^2 \times \log(m))$
	OUTLIERS[e]	$O(n^2 \times m^2)$	$O(n^2 \times m^2)$
		$\boldsymbol{O(n^2 \times m^2)}$	$\boldsymbol{O(n^2 \times m^2 \times \log(m))}$

3. Estimation of the Outlier Probability of Each Element

In the previous section, a theoretical framework was defined by expanding [1, 7], based on which the FIND_OUTLIER_REGION algorithm was constructed. This algorithm enables us to calculate all outlier regions for each element of the universe, and the complexity of this algorithm is almost linear. Ultimately, these results enable us to develop the solution proposed in this study (Figure 2): a computationally viable algorithm, valid for environments of large volumes of data, able to provide the outlier probability of each element of the universe. This algorithm was termed the *βμ_PROB algorithm*. Following a pattern similar to that followed in the previous section, first, a theoretical framework will be developed by expanding [1, 7], which will provide the mathematical tools we need to build the solution. Subsequently, the spatial and temporal complexity of the algorithm will be analysed.

3.1. Theoretical Framework: Probabilistic βμ Method. As mentioned above, the results from the previous section enable us to assess, for each $e \in U$, the region of β and μ values in which such element is an outlier. Let us call OUTLIERS_e the region found for a given element, $e \in U$.

Considering β and μ two random variables, let us call $\varphi(\beta, \mu)$ the probability density function of the random vector (β, μ). Then, the distribution function of (β, μ) would be

$$P(\beta \leq i, \mu \leq j) = \int_{-\infty}^{i} \int_{-\infty}^{j} \varphi(\beta, \mu) d\beta d\mu. \quad (6)$$

$\beta\mu$_PROB (U, X, R, PDF()): P

	Pseudo-code	***Comments***
1	OUTLIERS = **FIND_OUTLIER_REGION** (U, X, R)	*Apply probability distribution PDF for each region*
2	**for each** e ∈ U	*For every element of the universe*
	P[e] = 0	*Initial probability*
3	**for each** rect ∈ OUTLIERS[e]	*For each rectangle of exceptionality*
4	P[e] = P[e] + PDF(rect)	*Accumulate the probability of each rectangle*
5	**return** P	*Return P*

Algorithm 4: Pseudo-code of the $\beta\mu$_PROB algorithm.

Then, the probability that we are interested in calculating, P_e, that is, the probability that $e \in U$ is an outlier knowing OUTLIERS_e can be calculated from (6) using the following formula:

$$P_e = P((\beta, \mu) \in \text{OUTLIERS}_e) = \int_{\text{OUTLIERS}_e} \varphi(\beta, \mu) d\beta d\mu. \quad (7)$$

Considering that e is an outlier of β and μ values belonging to OUTLIERS_e.

Because β and μ are two independent random variables, then: $\varphi(\beta, \mu) = f(\beta)g(\mu)$, where $f(\beta)$ and $g(\mu)$ are the probability density functions of β and μ, respectively. Therefore,

$$P_e = \int_{\text{OUTLIERS}_e} f(\beta) \cdot g(\mu) d\beta d\mu. \quad (8)$$

We only have to replace the probability density functions of the parameters β and μ in (8) to calculate P_e and then calculate the resulting integral. In practice, most commonly, no information about the distribution of the parameters β and μ is available. Therefore, they will be both assumed to be uniformly distributed. If, in any context, this distribution is different from the expected, it is sufficient to calculate P_e with new functions, using some numerical method to calculate the integral if necessary. Based on this assumption, the resulting integral is easily calculated. Because $0 \le \beta < 0.5$ and $0 \le \mu \le 1$, based on the *Uniformity hypothesis* for the values of these thresholds, its probability density function would be

$$f(\beta) = \frac{1}{0.5 - 0} = 2, \qquad g(\mu) = \frac{1}{1 - 0} = 1. \quad (9)$$

Replacing these values in (8), we have

$$P_e = 2 \int_{\text{OUTLIERS}_e} d\beta d\mu. \quad (10)$$

And because $\int_{OUTLIERS_e} d\beta d\mu$ is the area of the OUTLIERS_e region,

$$P_e = 2\text{Area}(\text{OUTLIERS}_e). \quad (11)$$

This result may be interpreted as

$$P_e = \frac{\text{Area}(\text{OUTLIERS}_e)}{0.5}. \quad (12)$$

This is precisely the quotient between the area of the favourable region (the region of values (β, μ) for which e is an outlier) and the total area (the rectangle that defines the domain of the values (β, μ) on the plane).

3.2. Computational Implementation: $\beta\mu$_PROB Algorithm. The *Bμ_PROB* algorithm input consists of the following: a universe U (dataset) $|U| = n$, a concept $X \subseteq U \neq \varnothing$, equivalence relations $R = \{r_1, r_2, \ldots, r_n\}$, and a probability distribution function *PDF*(). Its output consists of estimating the probability P for each element of U in terms of their outlier status in the universe. Because the *FIND_OUTLIER_REGION* algorithm calculates the outlier region OUTLIERS, the probability is calculated using the formula shown in (12). A description in pseudo-code of the algorithm that implements the aforementioned aspects is presented in Algorithm 4.

The temporal complexity of the $\beta\mu$_PROB algorithm is affected by the temporal complexity of the process for determining the outlier region:

(i) Cost of determining the outlier region: temporal complexity *FIND-OUTLIER-REGION:* $\mathbf{O}(n^2 \times \text{m}^2 \times \log(m))$

(ii) Cost of determining the probability: (dataset) × (total number of rectangles region β-μ) = $(n) \times (n \times m^2) \rightarrow \mathbf{O}(n^2 \times m^2)$

Therefore, the temporal complexity of the algorithm $\beta\mu$_PROB, in the worst case, is $\mathbf{O}(n^2 \times m^2 \times \log(m))$.

The $\beta\mu$_PROB algorithm solves two key problems: the lack of a specific algorithm to perform this calculation and the complexity of the calculation performed by combining existing algorithms [1, 7]; the resultant reduction

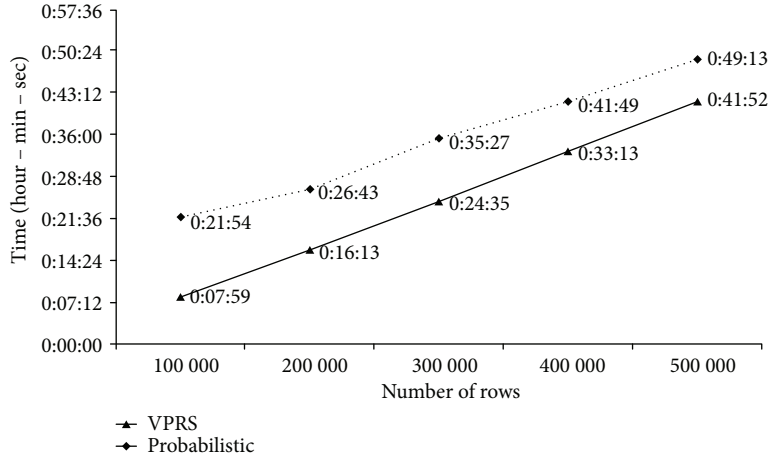

Figure 5: Comparison of run-times between the *VPRS* and $\beta\mu$*_PROB* algorithms.

in complexity allows application of the algorithm to environments with large volumes of information.

4. Validation of the Results

The algorithm validation tests have primarily focused on two aspects: comparing its run-times to those of the VPRS algorithm to obtain a realistic reference and assessing the detection quality of the $\beta\mu$*_PROB* algorithm. For such purposes, automatically generated random datasets and real-world datasets were used. Although performing quantitative comparisons to all algorithms identified in the state of the art is usually senseless due to the different nature of their application and usefulness, a comparison that allows us to contextualise each of them can be very interesting. Accordingly, the rest of the section is structured as follows: (1) evaluation of the algorithm run-times and comparison to the VPRS case, (2) evaluation of the detection quality, which is also compared to that of the VPRS, and (3) comparison of all RS-based methods to algorithms based on conventional methods and comparison to the advantages and drawbacks of each RS-based method of the study.

4.1. Run-Time Study. The *Bμ_PROB* algorithm run-time validation tests—compared to the *VPRS* algorithm [10]—are performed with large datasets having high dimensionality. Because similar results have been found in all the experiments, in this study, we show a specific example that is fully representative: multivariate synthetic data (random dataset automatically generated using statistical techniques that ensure a uniform distribution, among other aspects) with categorical and continuous attributes, with 500,000 records and with 100 columns. The number of equivalence relations covered is 100. The computing device used has the following characteristics: Intel(R) Core(TM)2 Quad processor CPU Q6600 @ 2.40 GHz, with 3.25 GB of memory running the Windows 7 Ultimate operating system.

Figure 5 shows the run-times assessed both for the $\beta\mu$*_PROB* and the *VPRS* algorithms. The equivalence relations and the number of columns remain fixed for the comparison, varying the number of records.

The curves show that both algorithms behave similarly—regarding the run-time—and that they are computationally efficient when analysing a large dataset with high dimensionality. Furthermore, the run-times are linear and advantageously require no preset thresholds.

This finding shows that although the order of temporal complexity for the BM_PROB algorithm is quadratic in the worst case, it may reach an almost linear order of temporal complexity when analysing datasets that are normally distributed.

4.2. Detection Quality Validation. Again, all experiments conducted yielded similar results; therefore, in this study, one of them is shown as a representative example. In this case, the dataset used was the Arrhythmia Data Set (data of patients with cardiovascular problems) from the UCI Machine Learning Data Repository [12]. These are multivariate data with real, complete, and categorical attributes. Here, 452 records from 279 fields were employed. The computing device used was an Intel(R) Core(TM) 2 Duo, CPU T5450 @ 1.66 GHz (with 2 CPUs), and 2046 MB of RAM running Windows Vista.

The concept $\boldsymbol{C}$ defined people with weight ≤40 kg, that is, low-weight people, and the following equivalence relations R:

(i) r_1: was established from the attribute heart rate: mean number of heart beats per minute of each person. The equivalence relation partitions the dataset into two equivalence classes: [44, 61] and [62, 163]

(ii) r_2: was established from the attribute number of intrinsic deflections: number of arterial bypasses of each person. The equivalence relation partitions the dataset into two equivalence classes: [0, 59] and [60, 100]

(iii) r_3: was established from the attribute height: height of a person expressed in centimetres. The equivalence relation partitions the dataset into two equivalence classes: [60, 175] and [176, 190]

Here, 12 outliers with contradictory values for low-weight people were intentionally injected into the dataset. The normal values of the attributes considered in the equivalence relations for low-weight people are as follows: heart rate >65, intrinsic deflections <50, and height <170 cm. Table 2 describes the outliers injected. The values in bold and italics represent contradictory values.

In the test, the following μ values were analysed: 0.2, 0.4, 0.6, 0.8, and 1. For each μ value, β was varied according to the following sequence of values: 0, 0.1, 0.2, and 0.3. The values 0.4 and 0.5 are not mentioned because the number of outliers detected remained 0 beyond $\beta = 0.3$. After applying the $\beta\mu$_PROB algorithm, different subsets formed by k elements, with $k \in (5, 10, 15, 20)$, are taken from the dataset with the highest outlier probability. Then, the number of injected outliers found in each of these subsets is analysed. Figure 6 shows the results achieved on this occasion.

The number of most likely elements (k) considered in each case shows that when $k = 5$, the 5 elements with the highest outlier probability are the 5 most contradictory elements of the dataset; when $k = 10$, the 10 elements with the highest outlier probability introduced in the dataset and, when $k = 15$ and $k = 20$, the 12 outliers intentionally injected already appeared among the most likely k. In summary, the 12 injected elements were always found among those with the highest outlier probability after applying the $\beta\mu$_PROB algorithm.

Table 3 presents the probability values determined using the $\beta\mu$_PROB algorithm for outliers injected into the dataset.

4.3. Comparison of the Outlier Detection Algorithms. Most outlier detection techniques and algorithms analysed are designed, to a greater or lesser extent, to solve a specific type of problem, even in a specific case. Valid comparisons between these algorithms are difficult to perform because they will considerably depend on the search target. However, it is interesting to perform a comparative study of the different existing methods highlighting the advantages from the current proposal in its field—the unsupervised provision of general results regarding all elements of the data universe by establishing specific initial conditions: concept and equivalence relations. Considering the above, Table 4 details how the $\beta\mu$_PROB algorithm may help to overcome the limitations of the methods studied when requiring generalisation.

The main advantage of RS-based proposals and, particularly, of the $\beta\mu$_PROB algorithm relative to conventional methods lies in its generalist character. Unsurprisingly, an algorithm specially designed to detect a specific type of outliers is usually better, both in terms of detection quality and spatial and temporal complexity. However, having a generic algorithm that is capable of addressing different types of problems, with different types of data, and able to behave reasonably with large volumes of data is a very interesting option that avoids having to design different algorithms each time new problems emerge or when the conditions of previously solved problems change.

Table 2: Outliers injected into the test dataset.

ID	Weight (kg)	Heart rate	# intrinsic deflections	Height (cm)
1	15	**60**	17	**180**
2	31	93	**68**	**178**
3	39	**50**	**82**	130
4	10	**53**	16	**188**
5	19	**45**	**90**	**190**
6	20	**48**	**86**	**183**
7	25	**50**	**71**	**180**
8	29	**55**	**75**	**179**
9	33	90	**60**	**176**
10	40	**61**	20	**186**
11	26	**50**	**99**	**180**
12	38	92	**100**	**178**

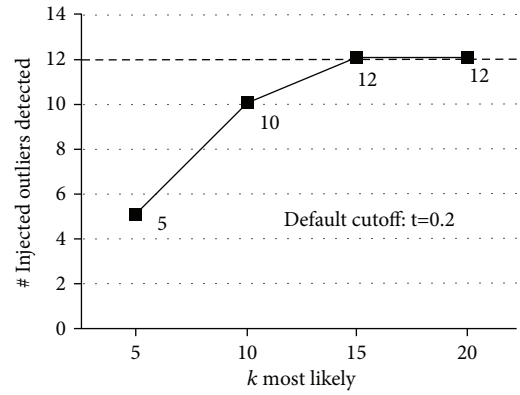

Figure 6: Number of injected outliers found between the k elements with the highest outlier probability.

After comparing algorithms based on conventional techniques and algorithms based on the RS model, a summary of the comparative study conducted between different RS algorithms and the proposed $\beta\mu$_PROB algorithm is presented in Table 5, outlining the advantages and disadvantages of each algorithm and highlighting the usefulness of the proposed algorithm.

5. Conclusions

Whereas VPRS has been applied to problems in multiple fields [13–16], particularly in the field of statistics [17], this study aimed to develop a new application of this model to the outlier detection problem, breaking with the traditional scheme followed by most existing detection methods. By defining the desired concept and equivalence relations, the algorithm provides unsupervised—and without needing to define neither the outlier threshold nor the classification error, which are both dependent on the problem—general results regarding all elements of the dataset. More specifically, it provides the outlier probability of each element from such universe. Therefore, this result is transcendent and

Table 3: Outlier probability of the 12 elements injected into the dataset.

ID	Weight (Kg)	Heart rate	# of intrinsic deflections	Height (cm)	Outlier probability
1	15	**60**	17	**180**	0.61884
2	31	93	**68**	**178**	0.7557252
3	39	**50**	**82**	130	0.6151009
4	10	**53**	16	**188**	0.61884
5	19	**45**	**90**	**190**	**0.8779342**
6	20	**48**	**86**	**183**	**0.8779342**
7	25	**50**	**71**	**180**	**0.8779342**
8	29	**55**	**75**	**179**	**0.8779342**
9	33	90	**60**	**176**	0.7557252
10	40	**61**	20	**186**	0.61884
11	26	**50**	**99**	**180**	**0.8779342**
12	38	92	**100**	**178**	0.7557252

Table 4: Characteristics of the RS-based methods compared to the limitations of conventional methods.

Comparison to STATISTICAL and DISTANCE-BASED METHODS
(i) Applicability to datasets with a mixture of continuous and discrete attributes. Equivalence relationships are a natural way to discretise continuous data.
(ii) Neither knowing the data distribution nor establishing data *distance* criteria is required.
(iii) Specifically, for $\beta = 0$, the quadratic temporal complexity problem of most *distance*-based methods is solved.
(iv) The dimensionality and dataset size do not limit the execution of the algorithms.
Comparison to DENSITY- and DEPTH-BASED METHODS
(i) There is no need to establish data density criteria in the dataset.
(ii) The dimensionality of the dataset does not limit the execution of the algorithm.
(iii) No time-consuming calculations are necessary, including calculating the *convex wrap*, which is required in most *depth*-based methods.
(iv) *FIND_OUTLIER_REGION* and $\beta\mu$*_PROB* provide unsupervised results without requiring the user to preset, before running the algorithm, the value of specific analysis parameters, which is necessary in *density*-based methods, such as *DBSCAN*.
(v) *Pawlak rough sets* and *VPRS* improve the temporal complexity compared to *depth*-based methods.
Comparison to METHODS BASED ON NEURAL NETWORKS
(i) No time-consuming processes must be previously established, for example, network training, required in some neuronal network models to ensure their learning.
(ii) The dimensionality of the dataset does not limit the execution of the algorithms.
(iii) The functionality of the algorithms does not depend on data *density* criteria, in contrast to some supervised models.
(iv) There is no need to model the data *distribution*, in contrast to some supervised models.
(v) Some approaches based on supervised networks establish the use of thresholds for various purposes in the *outlier* detection process. This is solved in the concept of the *FIND_OUTLIER_REGION* and $\beta\mu$*_PROB* algorithms.
Comparison to GENERAL OUTLIER DETECTION METHODS
(i) In contrast to most detection methods, which require successive executions of the algorithm until obtaining the set of outliers that actually meets the analysis criteria, β. PROB algorithm performs the single-run, unsupervised determination of the outlier probability of each element form a specific universe of data.

original because it paves the way for the analysis and solution of other particular problems. It allows us to have an overview of the data and thus to test its representativeness.

The algorithms presented demonstrate the computational feasibility of the proposed methods. Furthermore, they provide efficient computational solutions—in terms of temporal and spatial complexity—to the problems for which they were conceived.

The method proposed solved, in addition, other limitations of several detection methods: it may be applied to datasets with a mixture of types of attributes (continuous and discrete); its application requires no prior knowledge about the data distribution; within the scope of its application, the size and dimensionality of the dataset do not limit its correct operation; and no distance or density criteria must be established for the dataset to apply this algorithm.

Table 5: Comparative table of RS-based algorithms.

Advantages	Disadvantages
Pawlak rough sets algorithm	
(i) Shows the computational viability of the *Pawlak rough sets*-based detection method. (ii) Linear temporal and spatial lineal complexity regarding the cardinality of the dataset.	(i) DETERMINISTIC classification. (ii) The user must define the *outlier threshold*.
VPRS algorithm	
(i) Shows the computational viability of the *VPRS*-based detection method. (ii) Linear temporal and spatial lineal complexity regarding the cardinality of the dataset. (iii) NONDETERMINISTIC classification.	(i) The user must define the *outlier threshold* and the *classification error*. (ii) An inadequate selection of the *error* may lead to unsatisfactory results. Requires sufficient knowledge of specific aspects of the *dataset*.
FIND_OUTLIER_REGION algorithm	
(i) Shows the computational viability of the $\beta\mu$ Method. (ii) Maintains the nondeterminism of *VPRS*. (iii) Any specific result that could be obtained with the *Pawlak rough sets* and *VPRS* algorithms can be determined from the result obtained. (iv) The obtained region allows us to establish a stochastic approach to solving the problem of determining the outlier probability of a given element from a given dataset. (v) Its use is especially feasible when needing to determine the *outlier* condition of the elements of the *dataset* for a given set of threshold values.	(i) Temporal complexity: $O(n^2 \times m^2 \times \log(m))$ in the worst case. (ii) Spatial complexity: $O(n^2 \times m^2)$ in the worst case.
$\beta\mu$*_PROB algorithm*	
(i) Shows the computational viability of the method defined. (ii) Maintains the nondeterminism of *VPRS*. (iii) Has the same advantages as the FIND_OUTLIER _REGION algorithm. (iv) The user does not need to define the outlier threshold, the allowed classification error, or other criteria, such as distance or density. (v) No specific knowledge of the dataset is required, such as its distribution. (vi) The result obtained is more general than that obtained with *Pawlak rough sets* and *VPRS*. (vii) Is valid for datasets with mixed types of attributes (continuous and discrete).	(i) Temporal complexity: $O(n^2 \times m^2 \times \log(m))$ in the worst case. (ii) Spatial complexity: $O(n^2 \times m^2)$ in the worst case.

The results reported in the present study are the beginning of an in-depth study in the context of the general problem of outlier detection based on the RS model. Therefore, several problems that have not yet been solved may be identified and may be the next objectives of this on-going study. Accordingly, the following objectives have been identified: (a) to further improve the run-time of the algorithms by creating a distributed execution mechanism to use the computational power of several machines in one domain. In the current version of the algorithms, the user has to execute them on a single personal computer (PC), and (b) in the current version of the $\beta\mu$_PROB algorithm, the β threshold domain is [0; 0,5]. However, the establishment of a new upper bound could allow us to gain precision in the probability calculation, especially in the case of very contradictory elements for few β values. Accordingly, the BM/probabilistic algorithm should be modified to automatically determine the most appropriate value for a given level.

Data Availability

The main dataset used to support the findings of this study is public and you can access it in Maching Learning Repository: Arrhythmia Data Set at URL https://archive.ics.uci.edu/ml/datasets/arrhythmia.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Funding

The authors received Fund no. TIN2016-78103-C2-2-R.

Acknowledgments

This work has been supported by University of Alicante projects GRE14-02 and Smart University.

References

[1] Z. Pawlak, "Rough sets," *International Journal of Computer & Information Sciences*, vol. 11, no. 5, pp. 341–356, 1982.

[2] S. W. Han and J.-Y. Kim, "Rough set-based decision tree using the core attributes concept," in *Second International Conference on Innovative Computing, Informatio and Control (ICICIC 2007)*, p. 298, Kumamoto, Japan, 2007, IEEE Computer Society.

[3] C. Cheng, Y. Chen, and J. Chen, "Classifying initial returns of electronic Firm_s IPOs using entropy based rough sets in Taiwan trading systems," in *Second International Conference on Innovative Computing, Informatio and Control (ICICIC 2007)*, p. 82, Kumamoto, Japan, 2007, IEEE Computer Society.

[4] M. Hirokane, H. Konishi, A. Miyamoto, and F. Nishimura, "Extraction of minimal decision algorithm using rough sets and genetic algorithm," *Systems and Computers in Japan*, vol. 38, no. 4, pp. 39–51, 2007.

[5] F. Jiang, Y. Sui, and C. Cao, "Outlier detection using rough set theory," in *Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing*, pp. 79–87, Springer, 2005.

[6] F. Maciá-Pérez, J. V. Berna-Martinez, A. Fernández Oliva, and M. A. Abreu Ortega, "Algorithm for the detection of outliers based on the theory of rough sets," *Decision Support Systems*, vol. 75, pp. 63–75, 2015.

[7] W. Ziarko, "Variable precision rough set model," *Journal of Computer and System Sciences*, vol. 46, no. 1, pp. 39–59, 1993.

[8] W. P. Ziarko, Ed., *Rough Sets, Fuzzy Sets and Knowledge Discovery*, Springer, 1994.

[9] W. Ziarko, "Probabilistic decision tables in the variable precision rough set model," *Computational Intelligence*, vol. 17, no. 3, pp. 593–603, 2001.

[10] A. Fernández Oliva, M. Abreu Ortega, M. C. Fernández Baizán, and F. Maciá Pérez, "Método de detección no determinista de outliers basado en el modelo de conjuntos aproximados de precisión variable," in *Jornadas para el Desarrollo de Grandes Aplicaciones de Red (JDARE'09)*, pp. 131–148, Alicante, España, 2009.

[11] Z. Pawlak, *Rough Sets: Theoretical Aspects of Reasoning about Data*, Springer, 1991.

[12] "UCI machine learning repository," May 2009, https://cml.ics.uci.edu.

[13] Z. T. Gong, B. Z. Sun, Y. B. Shao, D. G. Chen, and Q. He, "Variable precision rough set model based on general relation," in *Proceedings of 2004 International Conference on Machine Learning and Cybernetics (IEEE Cat. No.04EX826)*, pp. 2490–2494, Shanghai, China, 2004.

[14] M. J. Beynon and N. Driffield, "An illustration of variable precision rough sets model: an analysis of the findings of the UK monopolies and mergers commission," *Computers & Operations Research*, vol. 32, no. 7, pp. 1739–1759, 2005.

[15] C. T. Su and J. H. Hsu, "Precision parameter in the variable precision rough sets model: an application," *Omega*, vol. 34, no. 2, pp. 149–157, 2006.

[16] V. U. Maheswari, A. Siromoney, and K. M. Mehata, "The variable precision rough set model for web usage mining," in *Web Intelligence Research and Development*, vol. 2198 of Lecture Notes in Computer Science, pp. 520–524, Springer, Berlin, Heidelberg, 2001.

[17] W. Ziarko, "Decision making with probabilistic decision tables," in *International Workshop on Rough Sets, Fuzzy Sets, Data Mining, and Granular-Soft Computing (RSFDGrC' 99)*, vol. 1711 of Lecture Notes in Computer Science, pp. 463–471, Springer, Yamaguchi, Japan, 1999.